Cert4Prep
Practice Test Software

Amazon

SOA-C03

AWS Certified CloudOps Engineer - Associate

Version: 8.5

Questions & Answers DEMO PDF

(Preview content before you buy)

Check the full version using the link below

https://cert4prep.com/exam/soa-c03

Unlock Full Features

Stay Updated: 90 days of free exam updates
Zero Risk: 30-day money-back policy
Instant Access: Download right after purchase
Always Here: 24/7 customer support team

Question 1. (Multi Select)

A CloudOps engineer needs to control access to groups of Amazon EC2 instances using AWS Systems Manager Session Manager. Specific tags on the EC2 instances have already been added.

Which additional actions should the CloudOps engineer take to control access? (Select TWO.)

A: Attach an IAM policy to the users or groups that require access to the EC2 instances.
B: Attach an IAM role to control access to the EC2 instances.
C: Create a placement group for the EC2 instances and add a specific tag.
D: Create a service account and attach it to the EC2 instances that need to be controlled.
E: Create an IAM policy that grants access to any EC2 instances with a tag specified in the Condition element.

Answer: A, E

Explanation:
AWS Systems Manager Session Manager allows secure, auditable instance access without SSH keys or inbound ports. To control access based on instance tags, CloudOps best practices require two configurations:
Attach an IAM policy to users or groups granting ssm:StartSession, ssm:DescribeInstanceInformation, and ssm:DescribeSessions.
Include a Condition element in the IAM policy referencing instance tags, such as Condition: {"StringEquals": {"ssm:resourceTag/Environment": "Production"}}.
This ensures users can start sessions only with instances that have matching tags, providing fine-grained access control.
AWS CloudOps documentation under Security and Compliance states:
“Use IAM policies with resource tags in the Condition element to restrict which managed instances users can access using Session Manager.”
Options B and D incorrectly suggest attaching roles or service accounts that are not relevant to user-level access control. Option C (placement groups) pertains to networking and performance, not access management. Therefore, A and E together provide tag-based, least-privilege access as required.

Question 2. (Single Select)

A company's ecommerce application is running on Amazon EC2 instances that are behind an Application Load Balancer (ALB). The instances are in an Auto Scaling group. Customers report that the website is occasionally down. When the website is down, it returns an HTTP 500 (server error) status code to customer browsers.

The Auto Scaling group's health check is configured for EC2 status checks, and the instances appear healthy.

Which solution will resolve the problem?

A: Replace the ALB with a Network Load Balancer.
B: Add Elastic Load Balancing (ELB) health checks to the Auto Scaling group.
C: Update the target group configuration on the ALB. Enable session affinity (sticky sessions).
D: Install the Amazon CloudWatch agent on all instances. Configure the agent to reboot the instances.

Answer: B

Explanation:
In this scenario, the EC2 instances pass their EC2 status checks, indicating that the operating system is responsive. However, the application hosted on the instance is failing intermittently, returning HTTP 500 errors. This demonstrates a discrepancy between the instance-level health and the application-level health.
According to AWS CloudOps best practices under Monitoring, Logging, Analysis, Remediation and Performance Optimization (SOA-C03 Domain 1), Auto Scaling groups should incorporate Elastic Load Balancing (ELB) health checks instead of relying solely on EC2 status checks. The ELB health check probes the application endpoint (for example, HTTP or HTTPS target group health checks), ensuring that the application itself is functioning correctly.
When an instance fails an ELB health check, Amazon EC2 Auto Scaling will automatically mark the instance as unhealthy and replace it with a new one, ensuring continuous availability and performance optimization.
Extract from AWS CloudOps (SOA-C03) Study Guide – Domain 1:
"Implement monitoring and health checks using ALB and EC2 Auto Scaling integration. Application Load Balancer health checks allow Auto Scaling to terminate and replace instances that fail application-level health checks, ensuring consistent application performance."
Extract from AWS Auto Scaling Documentation:
"When you enable the ELB health check type for your Auto Scaling group, Amazon EC2 Auto Scaling considers both EC2 status checks and Elastic Load Balancing health checks to determine instance health. If an instance fails the ELB health check, it is automatically replaced."

Therefore, the correct answer is B, as it ensures proper application-level monitoring and remediation using ALB-integrated ELB health checks—a core CloudOps operational practice for proactive incident response and availability assurance.
Reference (AWS CloudOps Verified Source Extracts):
AWS Certified CloudOps Engineer – Associate (SOA-C03) Exam Guide: Domain 1 – Monitoring, Logging, and Remediation.
AWS Auto Scaling User Guide: Health checks for Auto Scaling instances (Elastic Load Balancing integration).
AWS Well-Architected Framework – Operational Excellence and Reliability Pillars.
AWS Elastic Load Balancing Developer Guide – Target group health checks and monitoring.

Question 3. (Single Select)

A CloudOps engineer creates an AWS CloudFormation template to define an application stack that can be deployed in multiple AWS Regions. The CloudOps engineer also creates an Amazon CloudWatch dashboard by using the AWS Management Console. Each deployment of the application requires its own CloudWatch dashboard.

How can the CloudOps engineer automate the creation of the CloudWatch dashboard each time the application is deployed?

A: Create a script by using the AWS CLI to run the aws cloudformation put-dashboard command with the name of the dashboard. Run the command each time a new CloudFormation stack is created.
B: Export the existing CloudWatch dashboard as JSON. Update the CloudFormation template to define an AWS::CloudWatch::Dashboard resource. Include the exported JSON in the resource's DashboardBody property.
C: Update the CloudFormation template to define an AWS::CloudWatch::Dashboard resource. Use the intrinsic Ref function to reference the ID of the existing CloudWatch dashboard.
D: Update the CloudFormation template to define an AWS::CloudWatch::Dashboard resource. Specify the name of the existing dashboard in the DashboardName property.

Answer: B

Explanation:
According to CloudOps automation and monitoring best practices, CloudWatch dashboards should be

provisioned as infrastructure-as-code (IaC) resources using AWS CloudFormation to ensure consistency, repeatability, and version control. AWS CloudFormation supports the AWS::CloudWatch::Dashboard resource, where the DashboardBody property accepts a JSON object describing widgets, metrics, and layout.

By exporting the existing dashboard configuration as JSON and embedding it into the CloudFormation template, every deployment of the application automatically creates its corresponding dashboard. This method aligns with the CloudOps requirement for automated deployment and operational visibility within the same stack lifecycle.

AWS documentation explicitly states:

“Use the AWS::CloudWatch::Dashboard resource to create a dashboard from your template. You can include the same JSON you use to define a dashboard in the console.”

Option A requires manual execution. Options C and D incorrectly reference or reuse existing dashboards, failing to produce unique, deployment-specific dashboards.

Question 4. (Single Select)

A user working in the Amazon EC2 console increased the size of an Amazon Elastic Block Store (Amazon EBS) volume attached to an Amazon EC2 Windows instance. The change is not reflected in the file system.

What should a CloudOps engineer do to resolve this issue?

A: Extend the file system with operating system-level tools to use the new storage capacity.
B: Reattach the EBS volume to the EC2 instance.
C: Reboot the EC2 instance that is attached to the EBS volume.
D: Take a snapshot of the EBS volume. Replace the original volume with a volume that is created from the snapshot.

Answer: A

Explanation:

When an Amazon EBS volume is resized, the new storage capacity is immediately available to the attached EC2 instance. However, EBS does not automatically extend the file system. The CloudOps engineer must manually extend the file system within the operating system to utilize the additional space.

AWS documentation for EC2 and EBS specifies:

“After you increase the size of an EBS volume, use file system–specific tools to extend the file system so that the operating system can use the new storage capacity.”
On Windows instances, this can be achieved through Disk Management or diskpart commands. On Linux systems, utilities such as growpart and resize2fs are used.
Options B and C do not modify file system metadata and are ineffective. Option D unnecessarily replaces the volume, which adds risk and downtime. Thus, Option A aligns with the Monitoring and Performance Optimization practices of AWS CloudOps by properly extending the file system to recognize the new capacity.

Question 5. (Single Select)

A company has a workload that is sending log data to Amazon CloudWatch Logs. One of the fields includes a measure of application latency. A CloudOps engineer needs to monitor the p90 statistic of this field over time.

What should the CloudOps engineer do to meet this requirement?

A: Create an Amazon CloudWatch Contributor Insights rule on the log data.
B: Create a metric filter on the log data.
C: Create a subscription filter on the log data.
D: Create an Amazon CloudWatch Application Insights rule for the workload.

Answer: B

Explanation:
To analyze and visualize custom statistics such as the p90 latency (90th percentile), a CloudWatch metric must be generated from the log data. The correct method is to create a metric filter that extracts the latency value from each log event and publishes it as a CloudWatch metric. Once the metric is published, percentile statistics (p90, p95, etc.) can be displayed in CloudWatch dashboards or alarms.
AWS documentation states:
“You can use metric filters to extract numerical fields from log events and publish them as metrics in CloudWatch. CloudWatch supports percentile statistics such as p90 and p95 for these metrics.”
Contributor Insights (Option A) is for analyzing frequent contributors, not numeric distributions. Subscription filters (Option C) are used for log streaming, and Application Insights (Option D) provides monitoring of application health but not custom p90 statistics. Hence, Option B is the CloudOps-aligned,

minimal-overhead solution for percentile latency monitoring.

Need more info? Check the link below:

https://cert4prep.com/exam/soa-c03

Thanks for Being a Valued Cert4Prep User!

Guaranteed Success Pass Every Exam with Cert4Prep.

Save $15 instantly with promo code

LEARN15

Sales: sales@cert4prep.com

Support: support@cert4prep.com